Autonomous Organization-Based Adaptive Information Systems

Eric Matson
Department of Computer Science and Engineering
Wright State University
Dayton, OH 45435
Department of Electrical and Computer Engineering and Computer Science
University of Cincinnati
Cincinnati, OH 45221
eric.matson@wright.edu

Scott DeLoach
Multi-agent and Cooperative Robotics Lab
Department of Computing and Information Sciences, Kansas State University
234 Nichols Hall, Manhattan, Kansas 66506, USA
sdeloach@cis.ksu.edu

Abstract-*On the field of battle, a continual flow of information is necessary to achieve information superiority. The required information potentially originates from numerous information producers. Unfortunately, due to dynamic and often dangerous conditions, these information producers can be incapacitated or destroyed. To be successful, a battlefield information system must provide a continuous flow of information and thus adaptability and fault tolerance are key features. In this paper, we show how we build adaptive and fault tolerance information systems using an organizational model. We introduce our organizational model and provide details pertaining to its use in the development of a battlefield information system.*

1. INTRODUCTION

An infosphere as envisioned by the DoD in such programs and documents as the Joint Battlespace Infosphere (JBI) [1], Network Centric Warfare (NCW)[2,3], and Joint Vision 2020 is an amazingly complex network of information producers, processors, and users. The goal of an infosphere is to provide for and allow for close coordination of battlefield capabilities of diverse and varied data sources [4]. Unfortunately, the probability of specific information and/or information sources being available, predictable, and timely is unknown. While the infosphere is tasked with providing persistent information objects, the infosphere, due to its lack of control over information sources, cannot guarantee that the current information is most recent or best available. Thus, these systems are, by nature of the infosphere, susceptible to loss of individual information sources, which can significantly impair the ability of the system to accomplish its goal. Most systems are currently designed to work with a limited set of information source/type configurations so that even when the information it needs to execute correctly is available, it is limited in reaching its goal by its own rigid information configuration requirements. To overcome this rigidity, we need systems that can adapt to a dynamic information environment. In our current research, we are developing the theories, techniques, and tools that will allow systems to change their configuration, or organization, in response to their environment.

Our approach to building such systems is based on a cooperative multiagent team. However, instead of relying on a set of predefined configurations, we provide the agents with information about their own capabilities and their role and relationships within their team, which we term a multiagent organization. Given the proper protocols and algorithms for team coordination, such teams have the ability to adapt their configuration, or organization, at runtime in order to respond to their dynamic environment.

In the information systems application area, a team consists of agents playing the roles of information processors, information producers and information sources. Information processor agents understand how to fuse particular types of information and raw data to create new information that is usable by specific users such as field commanders. Information producer agents represent the actual sources of raw data and information in the infosphere. Information source agents understand the information they are required to generate and how to interface with information producers to obtain the necessary raw data. However, a given information source agent may not know all the ways that a particular type of information can be produced. Therefore, a particular system many employ many information source agents to generate the same type of information. Some may generate the information more accurately while others may generate it more quickly or even may be able to derive it from different sources. The key to the process is being able to pick the right information source at the right time for the right task. This assignment is thus equivalent to organizing – choosing the appropriate multi-agent organization for a particular task. Additionally, if an information source is lost during the process, the team must be able to reorganize in the middle of its operation.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2005		**00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Autonomous Organization-Based Adaptive Information Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Wright State University,Department of Computer Science and Engineering,Dayton,OH,45435**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**8**	

Related Research

In reviewing other AIS models, there has been work in the area of AIS [8], some even containing the notion of organization [9]. Others have approached AIS from an intentional Multi-agent System (MAS) approach [10]. While these approaches are functional AIS systems, they lack the ability to reorganize and adapt if required due to environmental effects. We seek to prove that our approach yields benefits with the addition of the organization model and the ability to self organize and adapt to the necessary requirements of the surrounding environment.

Battlefield Information Systems

The goal of a battlefield information system is to provide the commander with both tactical and strategic intelligence. To accomplish this, various types of sensors are used to detect events and objects of interest. This sensor data is then be combined, or fused, with other sensor data to provide a commander with a more complete picture of the battlefield. Due to the nature of war, there is a high probability that some of these sensors will become disabled. However, when sensors are lost, their information is still required in order to provide the battlefield commander with a complete picture.

To provide required information, a battlefield information system must overcome the loss of sensors and sensor data, while continuing to provide continuous information flow. To accomplish this, the battlefield information system must detect sensor failure and adapt it's processing in a timely manner. Thus, a battlefield information system must be adaptive and efficient.

An Adaptive Information System (AIS) can modify its processing algorithms to the available sensor inputs to produce information at various levels of efficiency and effectiveness. Thus, an AIS selects the best available data and fuses it in an attempt to answer user queries.

To demonstrate our research on organization-based systems, we developed a battlefield information system simulation battlefield simulation based on the Battle of Khafji from the 1991 Gulf War [5, 6]. We combined multiagent systems with our model of team organization to construct a highly adaptive, yet efficient information system. Central to our research is the organizational model, which captures the structure used by a team of agents to answer information queries. Our organizational model provides the knowledge the team must possess to adapt to information source loss and forms the foundation upon which all team-based reasoning takes place. This model is based on team goals and valid organizational structures (the required roles and relationships). We are also currently researching suitable reasoning techniques to determine (1) when reorganization is necessary and (2) how best to achieve that reorganization.

In our Battle of Khafji simulation, we developed an organization-based AIS to answer queries that might be asked by a commander in the field. The goal of our research is to show that adaptive multi-agent organizations, which have the ability to reorganize as needed, can accomplish their goals more effectively and consistently systems with statically defined organizations.

We begin this paper by presenting our organizational model in Section 2 and how we used it to implement a battlefield AIS in Section 3. Finally, we discuss our conclusions in Section 4 and further work in Section 5.

2. AGENT ORGANIZATION MODEL

To implement teams of autonomous, heterogeneous agents, we created an organizational model, which defines and constrains the required elements of a stable, adaptable and versatile team [7]. While most people have an intuitive idea of what an organization is, there are no standard, formal definitions, from an agent oriented perspective. However, in most organizational research, organizations have typically been understood as including agents playing roles within a structure in order to satisfy a given set of goals. Our proposed organizational model (O) contains a structural model, a state model and a transition function.

$$O = \langle O_{structure}, O_{state,} O_{trans} \rangle$$

Figure 1 shows the combined structural and state models using standard UML notation. The structural model includes a set of goals (G) that the team is attempting to achieve, a set of roles (R) that must be played to attain those goals, a set of capabilities (C) required to play those roles, and a set of rules or laws (L) that constrain the organization. The model also contains static relations between roles and goals (achieves), roles and capabilities (requires), and individual roles (related). Each major component is discussed in detail below.

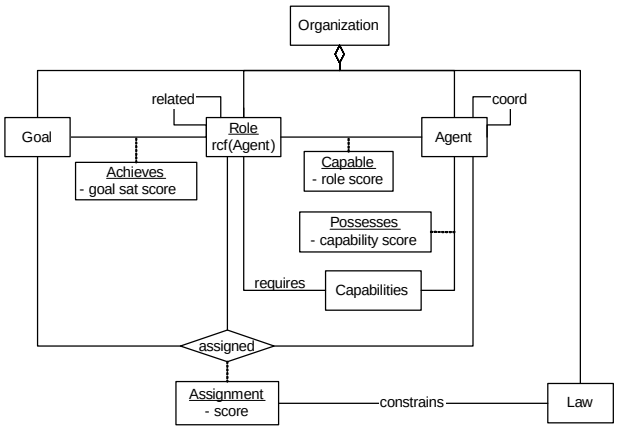

Figure 1. Organizational Model

Structural Model

Formally, we model the organization structure as a tuple.

$$O_{structure} = \langle G, R, L, C, achieves, related, requires \rangle$$

where

achieves: R, G → [0 .. 1]
related: R, R→ Boolean
requires: R, C → Boolean

The team *goals* include the goal definitions, goal-subgoal decomposition, and the relationship between the goals and their subgoals, which are either conjunctive or disjunctive. *Roles* define parts or positions that an agent may play in the team organization. In general, roles may be played by zero, one, or many agents simultaneously while agents may also play many roles at the same time. Each role requires a set of *capabilities*, which are inherent to particular agents and may include sensor capabilities, actuator capabilities, or computational capabilities. *Organizational rules* (or laws) are used to constrain the assignment of agents to roles and goals within the organization. Generic rules such as "an agent may only play one role at a time" or "agents may only work on a single goal at a time" are common. However, rules are often application specific, such as requiring particular agents to play specific roles. The structural model relations define mappings between the structural model components described above. A role that can be used to satisfy a particular goal is said to *achieve* that goal, while a role *requires* specific capabilities and may work directly with other roles, thus being *related* to those roles. Achieves is modeled as a function to capture the relative ability of a particular role to satisfy a given goal.

State Model

The organizational state model defines an instance of a team's organization and includes a set of agents (A) and the actual relationships between the agents and the various structural model components.

O_{state}= <A, possesses, capable, assigned, coord>

where

possesses: A, C → [0 .. 1]
capable: A, R → [0 .. 1]
assigned: A, R, G → [0 .. 1]
coord: A, A → Boolean

An agent that possesses the required capabilities for a particular role is said to be *capable* of playing that role. Since not all agents are created equally, possesses is modeled as a real valued function, where 0 represents no capability to play a role while 1 indicates an excellent capability. The capable function defines the ability of an agent to play a particular role and is computed based on the capabilities required to play that role.

Capability

Because capability is central to our organization, and its use in developing an AIS, we must define it more precisely before moving on. A capability's existence is based on the collective sense in which it is viewed. For an AIS, capability is defined upon the ability to create information of some level or type. To specify this we further define capabilities in relation to agent and roles that exist within a self-reorganizing multiagent team. As described above, an agent possesses specific capabilities while roles require particular capabilities.

The capability set of an agent, *Ca*, varies from a singleton set, if the agent possesses only one capability to the complete set of the capabilities that the agent intrinsically possesses. Normally even a simple agent has multiple capabilities.

$$C_a(a) = \{c \mid possesses(a,c)\}$$

Likewise, the capability set of a role, *Cr*, is the set of capabilities required to play that specific role. All roles must have at least one capability in order to accomplish some task or goal.

$$C_r(r) = \{c \mid requires(r,c)\}$$

An agent is capable of playing a role if $Cr(r) \subseteq Ca(a)$. How well agent a can play role r is determined by the *role capability function* (*rcf*) that is part of each role definition. The *rcf* is part of the role and defines a role-specific computation based on the capabilities possessed by an agent. If an agent does not possess one of the required capabilities, then the agent has no capacity to play that role and $r.rcf(a) = 0$. Thus, the *capability score* of an agent playing a particular role is defined by

$$capable(a,r) = r.rcf(a)$$

During the organization process, a specific agent is selected to play a particular role in order to satisfy a specific goal. This relationship is captured by the *assigned* function, which includes a real valued *score* that captures how well an agent, playing a specific role, can satisfy a given goal. This score is computed by

$$assigned(a,r,g).score = capable(a,r) * achieves(g,r)$$

When an agent is actually working directly with another agent, it is coordinating (*coord*) with that agent. Thus, the state model defines the current state of the team organization within the structure provided by the structural model.

Transition Function

The organization transition function defines how the organization may transition from one organizational state to another over the lifetime of the organization, $O_{state(n)} \rightarrow O_{state(n+1)}$. Since the team members (agents), as well as their individual capabilities, may change over time, this function cannot be predefined. It must be computed based on the current state, the goals that are still being pursued, and the organizational rules. In our present research with purely autonomous teams, we have only considered reorganization that involves the state of the organization. However, we have defined two distinct types of reorganization: state reorganization, which only allows the modification of the organization state, and structure reorganization, which allows modification of the organization structure (and may require state reorganization to keep the organization consistent). To

define state reorganization, we simply need to impose the restriction that

$$O_{trans}(O).O_{structure} = O.O_{structure}$$

Technically, this restriction only allows changes to the set of agents, A, the coord relation, and the possesses, capable, and assigned functions. However, not all these components are actually under the control of the organization. For our purposes, we assume that agents may enter or leave organizations or relationships, but that these actions are triggers that cause reorganizations are not the result of reorganizations. Likewise, possesses (and thus capable as well) is an automatic calculation on the part of an agent that determines the roles that it can play in the organization. This calculation is totally under control of the agent (i.e. the agent may lie) and the organization can only use this information in deciding its organizational structure. Changes in an agent's capabilities may also trigger reorganization. That leaves the two elements that can be modified via state reorganization: assigned and coord. Thus, we define state reorganization as:

$$O_{trans(state)} : O \rightarrow O$$

where

$$\begin{aligned} & O_{trans(state)}(O).\ O_{struct} = O.O_{struct} \\ & \quad \wedge\ O_{trans(state)}(O).\ O_{state}.A = O_{state}.A \\ & \quad \wedge\ O_{trans(state)}(O).O_{state}.\text{possesses} = O_{state}.\text{possesses} \\ & \quad \wedge\ O_{trans(state)}(O).O_{state}.\text{capable} = O_{state}.\text{capable} \end{aligned}$$

Reorganization Triggers

There are a variety of events that may occur in the lifecycle of a multiagent team that may require it to reorganize. In general, *reorganization* is initiated when an event occurs such that the team is no longer capable of reaching its overall goal, or when it realizes that it could reach its goal in a more efficient or effective manner. When the team is no longer capable of reaching its overall goal, we call this a *goal failure*. We have currently identified three role-related goal failure scenarios:

1. When a required role has not been assigned
2. When an agent relinquishes some required role
3. When an agent suffers a failure that keeps it from accomplishing its role

A team may also reorganize for efficiency or effectiveness. When a team reorganizes for *efficiency*, the team is still accomplishing its goals; it is just not doing it as efficiently as possible. In an information system we can equate efficiency to timeliness. Thus, if an information system has a requirement to produce new information by a set deadline, or consistently every few minutes, it may have to reorganize in order to meet those deadlines.

Reorganizing for *effectiveness* can be equated to the quality of information. In an intelligence gathering system, this is often quantified in terms of a confidence level. In this case, a commander might need a certain confidence level in a piece of information before making a decision and information timeliness may be able to be traded for quality. Quality failure can be detected based on the current system output (the current output does not meet quality requirements) or predicted in advance.

If we assume all efficiency and effectiveness goals are modeled as non-functional requirements, then the *capable* function captures all the data necessary to assess organizational effectiveness and efficiency. Triggering an efficiency or effectiveness based reorganization requires specific roles to monitor special conditions, such as those discussed above.

Reorganization

Once a triggering event occurs, the team must be capable of deciding whether reorganization is actually necessary. We currently assume that an organization always strives to operate in the optimal configuration. To achieve this optimal organization, the assignment of agents to roles and goals, must be maximized. If the organization has a choice in which agents play which roles, it should generally choose the more capable. Ideally, an organization will select the best set of assignments to maximize its ability to achieve its goals, which requires maximizing its organizational capability score, Os, given by

$$Os = \sum_{\forall a,r,g} assigned(a,r,g)$$

where *assigned(a,r,g) = 0* if that agent is not assigned to play a specific role to satisfy a goal.

We are currently using centralized search algorithms to find the optimal organization. However, we are investigating the use of distributed algorithms that can give acceptable capability scores without a complete reorganization, which is critical in showing that such an approach is scalable to hundreds or thousands of agents.

An illustrative example of the use of our organizational model is given in Figure 2. The boxes at the top of the diagram represent goals (A … G), the circles represent roles (R1 … R5), the pentagons represent capabilities (C1 … C5), and the rounded rectangles are agents (A1 … A4). The arcs under goals A and B denote that they are conjunctive goals (all subgoals must be achieved), whereas goal C is disjunctive (either F or G may be achieved to satisfy C).

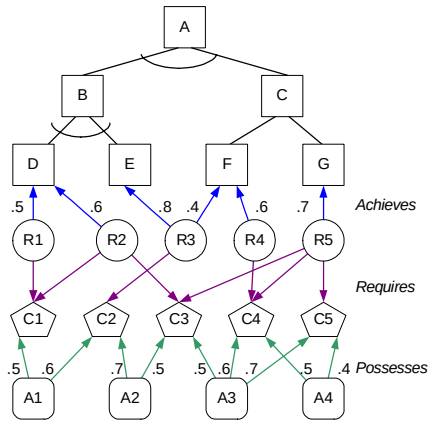

Figure 2. Organization Example

The arrows between the entities represent the *achieves*, *requires*, and *possesses* functions/relations as defined above. The numbers beside the arrows represent the function value (e.g., possesses(A1,C1) = 0.5). Given this example, we can compute the *capable* function value for each agent – role pair (assuming *rcf* for each role is simply an average of all required capabilities) as shown in Table 1.

Table 1. Capable Function

	R1	R2	R3	R4	R5
A1	0.5	0	0.6	0	0
A2	0	0	0.7	0	0
A3	0	0	0	0.6	0.6
A4	0	0	0	0.5	0

Combining the *capable* scores with the *achieves* score, we can compute the organizational capability score, Os, for any set of assignments that might be made. Although there are no agents capable of playing the R2 role, we can still find a viable organization. Assuming we make common assumptions that we can only assign a single agent to work on each goal and that only one of the disjunctive goals F and G can be active at any one time, we see that the optimum assignments (yielding the maximum organizational capability score) are as follows:

assigned(A1,R1,D) = 0.25
assigned(A2,R3,E) = 0.56
assigned(A3,R5,G) = 0.42
Os = 1.23

3. Adaptive Information System

A modern battlefield contains numerous heterogeneous data sensors whose data streams must be fused to create information. The fusion of data from these sensors provides the foundation for the information gathered and analyzed from the field of battle. The goal of information fusion is to elevate the battlefield command's understanding of adversary activities. Knowledge of the enemy's movements will be used to provide strategic advantage. To satisfy the requirements of the infosphere infrastructure, the functionality to fuse the data must be augmented with a fault tolerant mechanism to adapt to the dynamic conditions of the battlefield. To create an infrastructure that meets the requirements of continuous information flow, adaptability, fault tolerance, scalability and fusing of numerous heterogeneous data sources, we have extended a multi-agent system with a self-reorganizing architecture to create an autonomous, adaptive information system.

Design

Our organization-based AIS implementation results in a cooperative agent team with no pre-defined leaders, no subordinate role assignments, and no static hierarchical structure. In this model, air, satellite and ground-based sensors are monitored to evaluate enemy force deployment and movement intelligence. Each type of sensor has the capability to track and return a single, specific data stream.

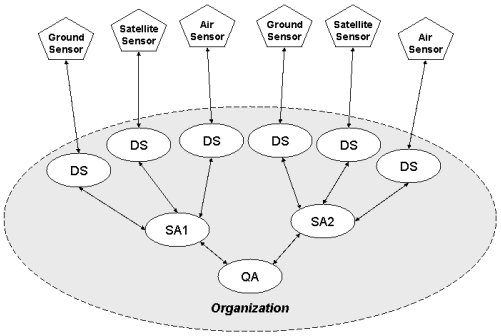

Figure 3. AIS Overview

We have defined three agent types for the implementation. Each fills a role in the situational analysis of the battlefield simulation scenarios. Figure 33 shows an example AIS configuration employing the defined agent types, which are data sensor agents, synthesis agents, and query agents.

Data Sensor Agents (DS) are the interface between the hardware sensors and the Synthesis Agents. Our current research utilizes satellite, air and ground based sensors. Each of the hardware sensors works with a DS to monitor its data and communicate to the organization. Some sensors have more than one capability, so the Data Sensor Agent capability models that of the physical sensor. For example a ground sensor may be limited to sensing heat blooms on M1-AI tanks, a satellite may possess the capability to sense a number of environmental factors.

Synthesis Agents (SA) have a defined capability set that allows them to work with a set of DS's. Its function is to fuse data from various sensor types to formulate answers to requests for information of the Query Agent.

Query Agents (QA) translate, manage and communicate the present query to the current Synthesis Agents. The queries have two forms: transient or persistent. A transient query is executed once and has a definite start and end. A persistent query is executed periodically for an indefinite period of time.

The organization consists of the collection of all Data Sensor Agents, Synthesis Agents and the Query Agent required for accomplishing the goal of resolving the query. Our current implementation allows for the team to self organize and work to satisfy a set of goals, which are defined by queries from the user. Once the team organizes, if it successfully satisfies its goals with no sensor loss, there is no reason to reorganize. However, if the team detects a sensor loss or failure, the team must evaluate the impact based on the current goals and team organization. The process of reorganization is triggered if the team can no longer satisfy its defined goal set with the current organization. During the reorganization process, the team discerns whether the capabilities possessed are sufficient to meet the capabilities required to reorganize, based on remaining goals.

Implementation

The implementation of our organization-based AIS model is developed using Java giving the AIS the ability to operate on all required platforms. There are three elements to the implemented AIS; the organization model, query interface and the battlefield simulator. All classes of the organization model have been implemented for this research. The query interface is a simplified way of entering queries to the organization. The battlefield simulator creates an environment to evaluate the organization over the life of a query.

When a query is submitted, it defines a new goal, or set of goals, for the organization. The organization is instantiated to begin query satisfaction. Our organization-based AIS uses two types of queries differing on their temporal constraints. Transient queries are singly executed with definite start and end points. Persistent queries are executed over an indefinite period of time with a definite start point but no stop point, at least defined at initiation.

Queries

When a query is launched there is no assurance the resources required to execute the query will be available for the duration. This applies to Transient queries but more so to Persistent queries because of the time required to complete execution. A Persistent query may execute and report information over a long time, heightening the probability that sensors will be destroyed or incapacitated that are being used in the information gathering to satisfy the information goals of the query. If a sensor is damaged, the current organization may be required to re-evaluate its global capability to satisfy the query requirements. If the requirements are not being met, a re-organization is triggered to re-instantiate the static organizational model into a new instance to meet the query goal requirements.

Queries are created using a structured natural language interface. The query is parsed and evaluated by the QA assigned by the organization. Examples of these queries are:

- *Show all tanks within sector 5.*
- *Show all vehicle movement within the sector.*

Once the QA has successfully parsed the query, goals are created and passed to the appropriate SAs with the capabilities to answer that specific query.

Simulator

To produce a realistic a simulation, the Battlefield Simulator, shown in Figure 4, was developed. The simulator contains a map of Iraq with the ability to simulate enemy operations in specific sectors. These enemy operations (troop movements, armor columns, etc.) are monitored by our AIS.

The simulator also generates the sensor simulation that provides the coupling and data for the Data Sensors. The air, ground and satellite sensor types are native to the simulator. When required, the sensor types are attached to the organization in response to the transient or persistent query. The simulator provides the ability to delete sensors simulating the sudden loss of sensor function in battle. The loss of the sensors allows the testing and validation of the organization based AIS.

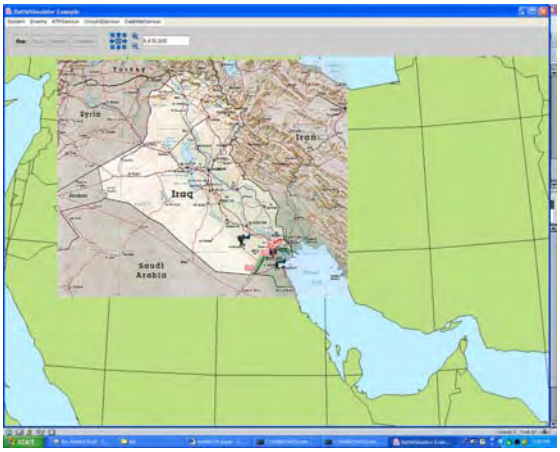

Figure 4. Battlefield Simulator

Example Scenario

This section presents a simulation scenario where our AIS is being used to provide fused information to aid in battlefield situation assessment. Persistent queries posed by the commander are answered based on the data generated by the Battlefield Simulator. Ground sensors detect the presence of troops, air sensors detect tanks within a sector, and satellite sensors detect helicopters within the sector. Since these are persistent queries, each sensor creates a constant stream of data describing the current battlefield.

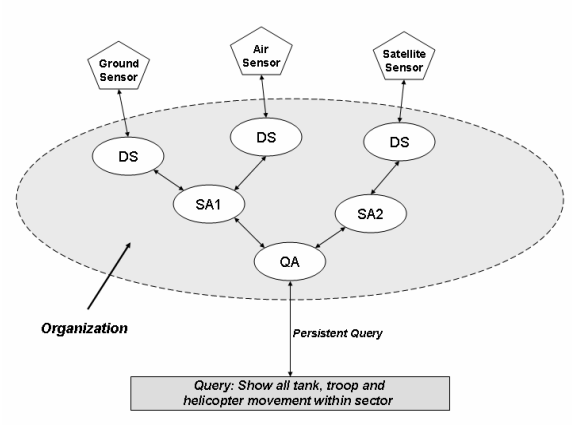

Figure 5. Persistent Query Organization

Figure 5 shows the initial organization set up to answer the persistent query, "Show all tank, troop and helicopter movement within sector". The organization requires the minimal capabilities of 3 sensors and 3 DS agents to interpret the raw data. SA1 possesses the capability to accept data from ground and air sensors and synthesize it for return to the QA. SA2 accepts and passes data from the satellite via the DS agent. The organization's assignment tuples will be:

assigned(A1,DS,Ground Sensor) = 1.00
assigned(A2,DS,Air Sensor) = 1.00
assigned(A3,DS,Satellite Sensor) = 1.00
assigned(A4,SA1,DS_{Ground} ^ DS_{Air}) = 0.65
assigned(A5,SA2,$DS_{Satellite}$) = 0.45
assigned(A6,QA,SA1 ^ SA2) = 0.39
Os = 4.49

The overall capability score to satisfy the query is 4.49. If a sensor is lost or incapacitated, the organization considers whether a re-organization is warranted. The loss of an agent within the organization also triggers an evaluation to determine if re-organization is required. In the Persistent Query Organization scenario shown in Figure 5, the loss of any sensor will trigger a re-organization. The change from $O_{state(current)} \rightarrow O_{state(current+1)}$ results in relationship changes as well as changes to elements of the organization state.

Figure 6 shows the same AIS after reorganizing due to the loss of the air sensor that was used to monitor the tanks on the battlefield. The loss of a physical sensor or the attached DS requires switching to another air sensor unit and allocating another DS to interpret the sensor's data. In this case, some organization relationships remain constant from from $O_{state(current)} \rightarrow O_{state(current+1)}$ such as QA, SA1 and SA2.

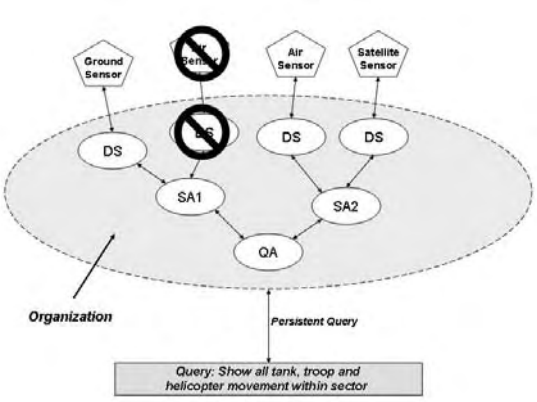

Figure 6. Persistent Query, Post Reorganization

In the process of reorganization, the capability evaluation determines SA2 now has a higher capability to synthesize data from the satellite and air sensors. This being the case, the sensor loss and ensuing reorganization resulted in an organization of higher capability – the organization score increased from 4.49 to 4.77. Obviously, if the new air sensor had been available when the original organization was formed, it would have been part of the optimal organization.

assigned(A1,DS,Ground Sensor) = 1.00
assigned(A7,DS,Air Sensor) = 1.00
assigned(A3,DS,Satellite Sensor) = 1.00
assigned(A4,SA1,DS_{Ground}) = 0.65
assigned(A5,SA2,$DS_{Satellite}$^ DS_{Air}) = 0.73
assigned(A6,QA,SA1 ^ SA2) = 0.39
Os = 4.77

Because of the potential duration of a persistent query and the conditions under which the sensors are subjected, it is possible that the organization may be transitioned many times over the course of a single query. The ability to continually evaluate the organization and reason about its viability is central to the nature of our organization-based AIS.

4. CONCLUSIONS

In this paper, we showed how we used our organizational model to create a flexible AIS capable of providing fault tolerance, adaptability and recovery on the battlefield. When an organization encounters a loss of capability, the organization evaluates and potentially responds with a reorganization to compensate for the lost organization member.

Our initial results show the ability to complete the processes of organization and reorganization within the structure of the AIS. The current implementation allows the organization to monitor itself and self reorganize in the event of the loss of an organization member agent. Furthermore, the organization uses the reorganization decision states to determine how best to satisfy its goals.

An advantage of a multi-agent system using the organization theoretic model is its extensibility. The

practical, numerical limits to the number of agents, roles or goals integrated and included in an organization is based upon system resources instead of limited by the organization model design.

In a previous evaluation, exhaustive organization and reorganization scenarios were executed to determine the success ratio of the ability of an instantiated organization to overcome the loss of a sensor [11]. Sensors were simulated using random number generating sequences that determine operational ability. When a sensor would go off-line, a re-organization was triggered and a new organization formed. All cases were successful in the ability to successfully re-organize due to our use of software-based sensors that could be generated as required. In a realistic battlefield information system, new hardware-based sensors are not so readily available.

5. FUTURE WORK

This work is part of a larger effort to more fully define the usefulness of an organizational theoretic approach to building a multi-agent system. In the near future, we plan to add new sensor types and thus assign more, different types of agent capabilities. This will allow us to more fully evaluate the scalability of the organizational model and the effectiveness of our organizational reasoning techniques.

Another goal is to investigate the use of effectiveness and efficiency as reorganization triggers. In a battlefield, information system effectiveness and efficiency generally refer to timeliness and confidence levels. If an organization has become inefficient, below a defined threshold, then the team can trigger a reorganization to improve efficiency. A similar stimulus and response relationship would exist for organizational effectiveness.

Finally, we plan to apply our organization theoretic approach to areas other than information systems. In particular, we are considering cooperative robotics, which has direct applicability to the control of uninhabited military vehicles.

6. ACKNOWLEDGEMENTS

This research is supported by a grant from the Air Force Office of Scientific Research (AFOSR) under grant number F49620-02-1-0427.

7. REFERENCES

[1] Air Force Research Labs, Joint Battlespace Infosphere, http://www.rl.af.mil/programs/jbi/default.cfm, 2004.

[2] R. Marmelstein. Force Templates: *A Blueprint for Coalition Interaction within an Infosphere*. IEEE Intelligent Systems, May/June 2002.

[3] J. Ladymon. *Network Centric Warfare and its Function in the Realm of Interoperability*. Acquisition Review Quarterly, Summer 2001, pp. 111-120.

[4] A. Cebrowski, (Vice Admiral, U.S. Navy), J. Garstka. *Network-Centric Warfare: Its Origin and Future*.http://www.usni.org/Proceedings/Articles98/PROcebrowski.htm, January 1998

[5] R. Grant. "*The Epic Little Battle of Khafji*," Air Force Magazine: Journal of the Air Force Association. Volume 81, No. 2, February 1998

[6] J.B. Rochelle. *The Battle of Khafji: Implications for Air Power*. Graduate Thesis, School of Advanced Airpower Studies, Air University, Maxwell Air Force Base, Alabama, 1997.

[7] E. Matson, S. DeLoach. *Organizational Model for Cooperative and Sustaining Robotic Ecologies*, Proceedings of the Robosphere 2002 Workshop, NASA Ames Research Center, Moffett Field, California, November 14-15, 2002.

[8] S. Garlatti, S. Iksal, et al., *Adaptive On-line Information System*, Information System, Analysis and Synthesis, (ISAS'99), Orlando, Florida 1050-1057, 1999.

[9] K. Ragab, T. Ono, N. Kaji, K. Mori. *Autonomous Decentralized Community Concept and Architecture for a Complex Adaptive Information System.* The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03). San Juan, Puerto Rico, May 28 - 30, 2003.

[10] A. Cardon, F. Lesage. *Toward Adaptive Information Systems: Considering Concern and Intentionality. The* Eleventh Workshop on Knowledge Acquisition, Modeling and Management (KAW'98). Banff, Alberta, Canada. April 18 -23, 1998.

[11] Eric Matson, Scott DeLoach. *Organization-based Adaptive Information Systems for Battlefield Situational Analysis*, Proceedings of the IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems (IEEE KIMAS '03), Boston, MA, October 1-3, 2003.